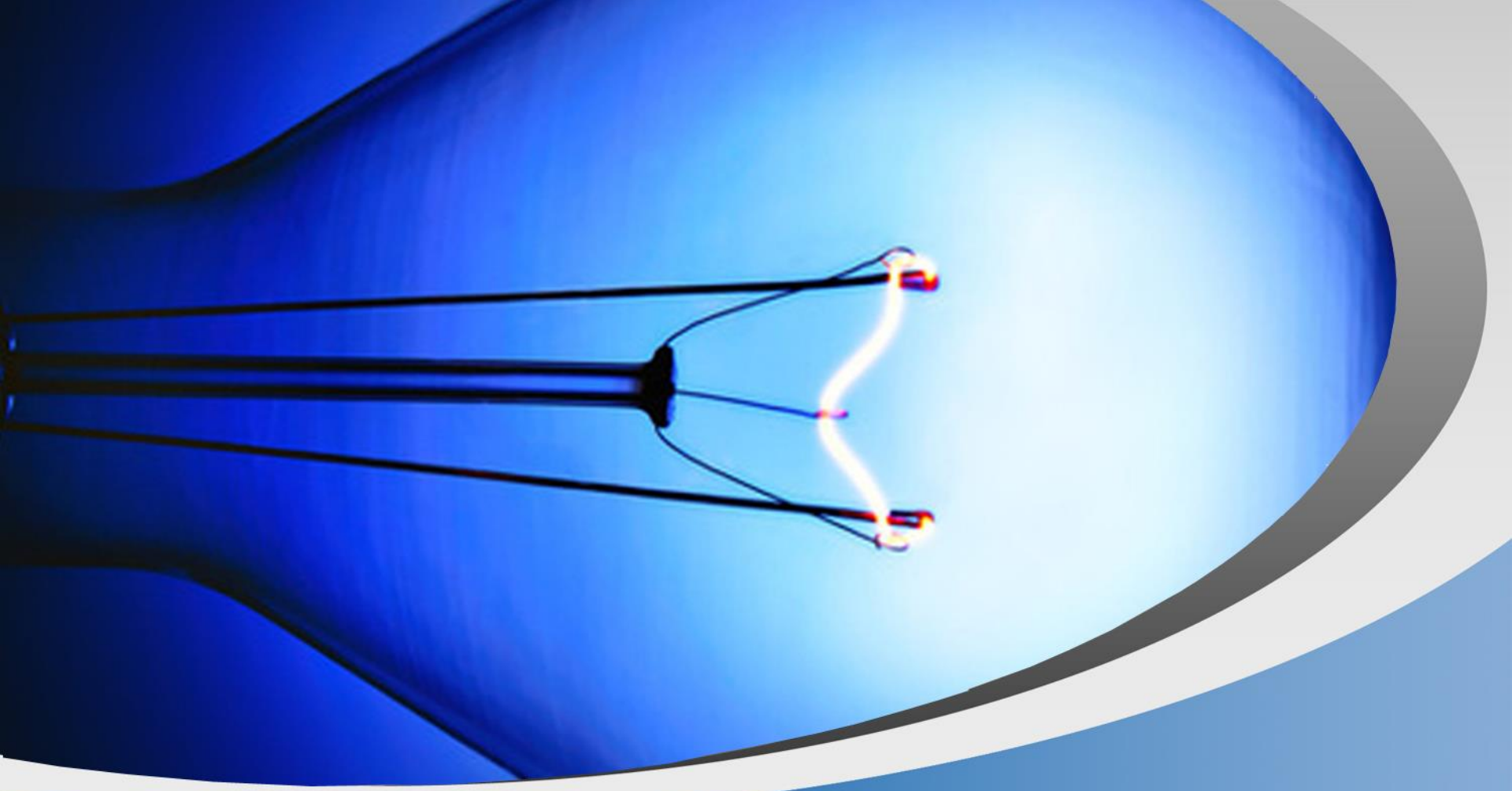


SystemC CCI WG
Inter Parameter Value Constraints

P V S Phaneendra, CircuitSutra Technologies Pvt. Ltd

August 2011

accellera SYSTEMS INITIATIVE

Inter_Parameter_Value_Constraints

- **Objective of the present example is to demonstrate the following :**
  - Establishing and maintaining inter-parameter value constraints (R16)
  - Reporting inter-parameter value constraint violations

- **The dynamic ability to reject values have been explained and contrasted to (static) locking**

Example Illustration

'X'	**<**	$2^n - 1$	
where,	***X***	:	Memory Block Size, and
	n	:	Number of address Lines

Condition to be tested

Example Illustration (1)

Get and store the handles for interrelated parameters of
address_lines_register and ***memory_block*** modules

Configurator

Defines a function that checks the condition : $X < 2^n - 1$

void TestCondition(int <addr_lines>, int <memory_size>) function *generates log* whether or not the condition specifed is met

Parameter Name : curr_addr_lines

Default Value : 9

address_lines_register

Parameter Name : mem_size

Default Value : 500

memory_block

processor

Example Illustration (2)

Get and store the handles for interrelated parameters of
address_lines_register and ***memory_block*** modules

Infrastructure

Configurator

•*Processor* module ***registers*** **POST_WRITE** callbacks on base parameters of the child modules
•Within callbacks implemention, the current values of the cci-parameters are fed to the ***TestCondition*** function

Parameter Name : curr_addr_lines

Default Value : 9

address_lines_register

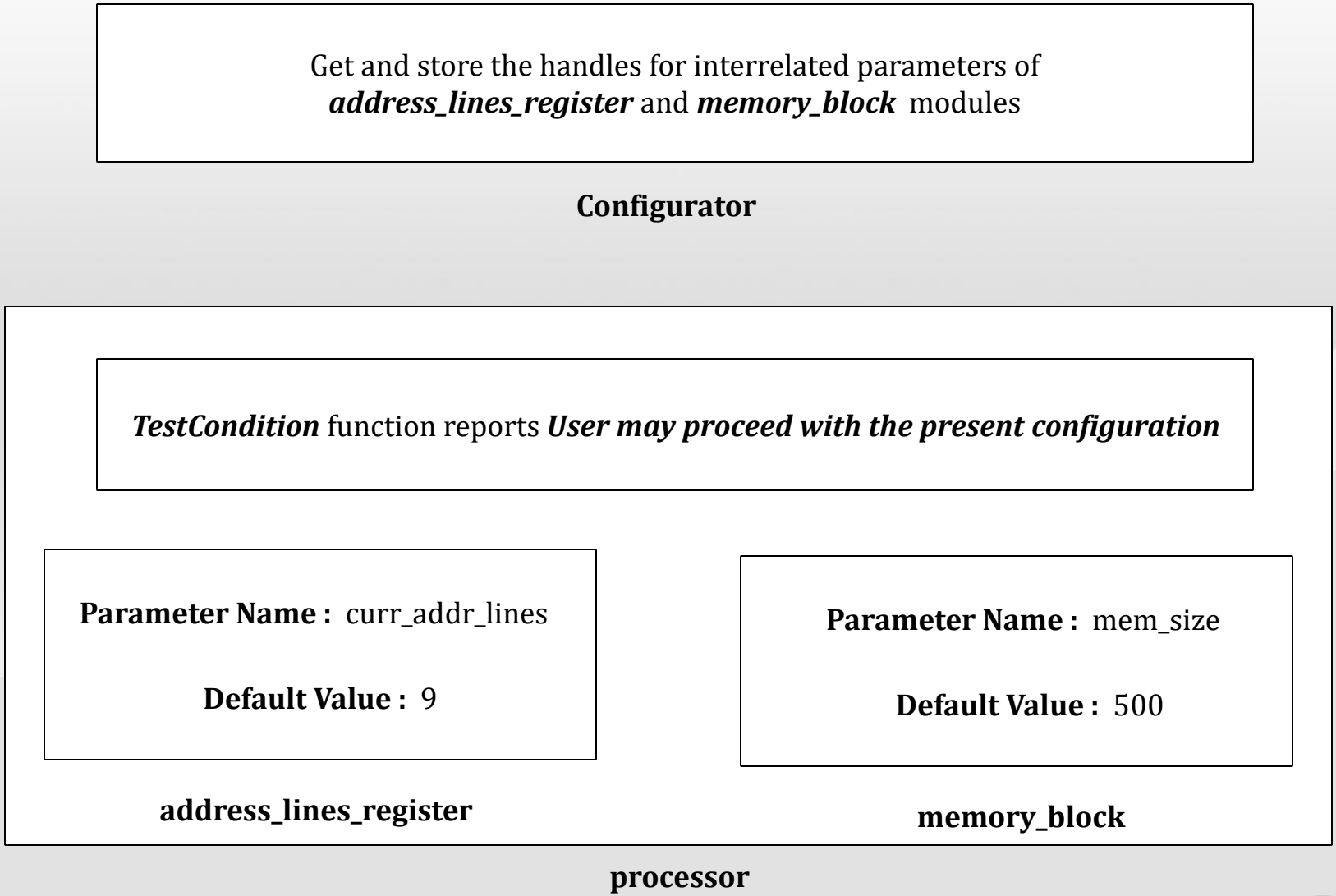
Parameter Name : mem_size

Default Value : 500

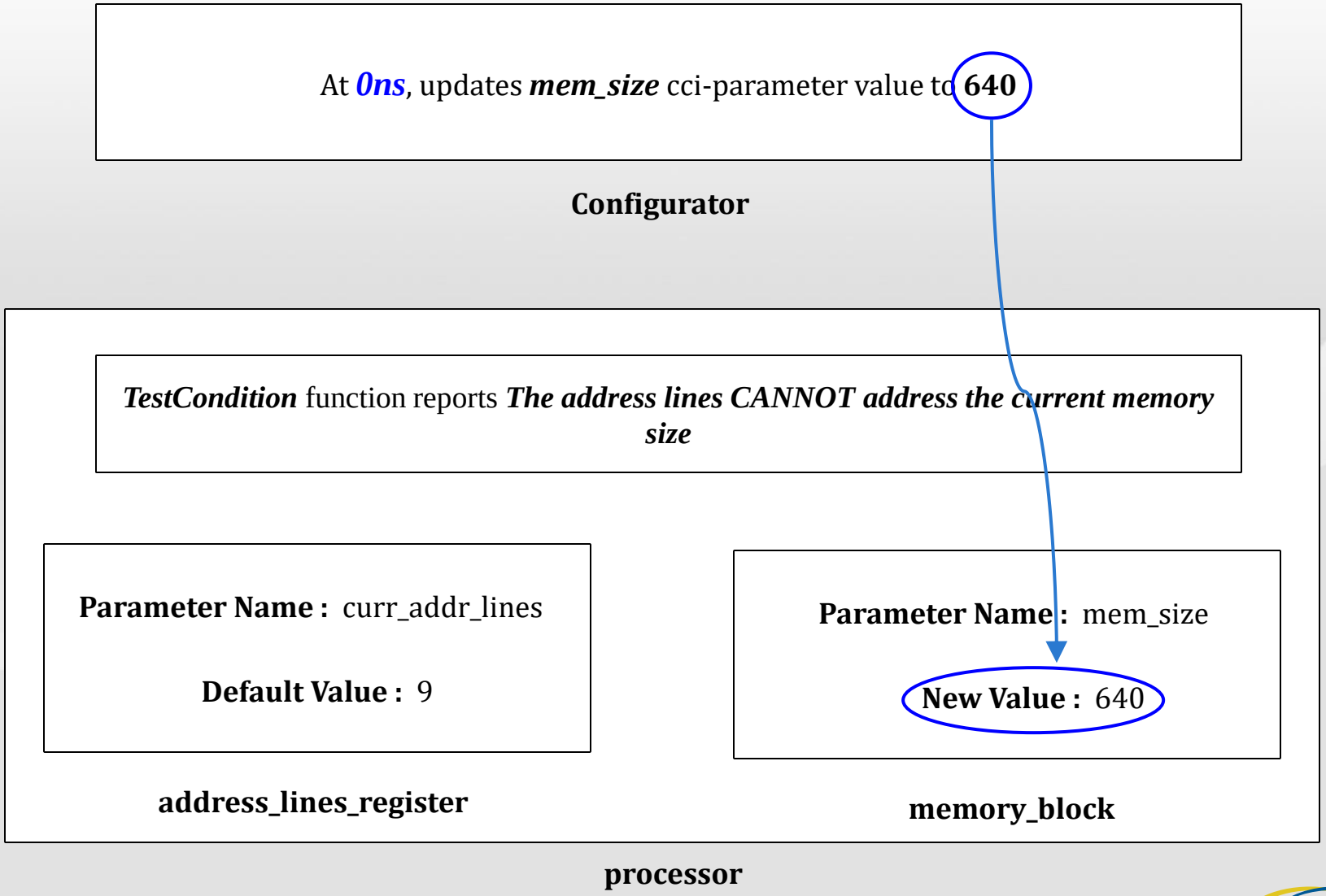
memory_block

processor

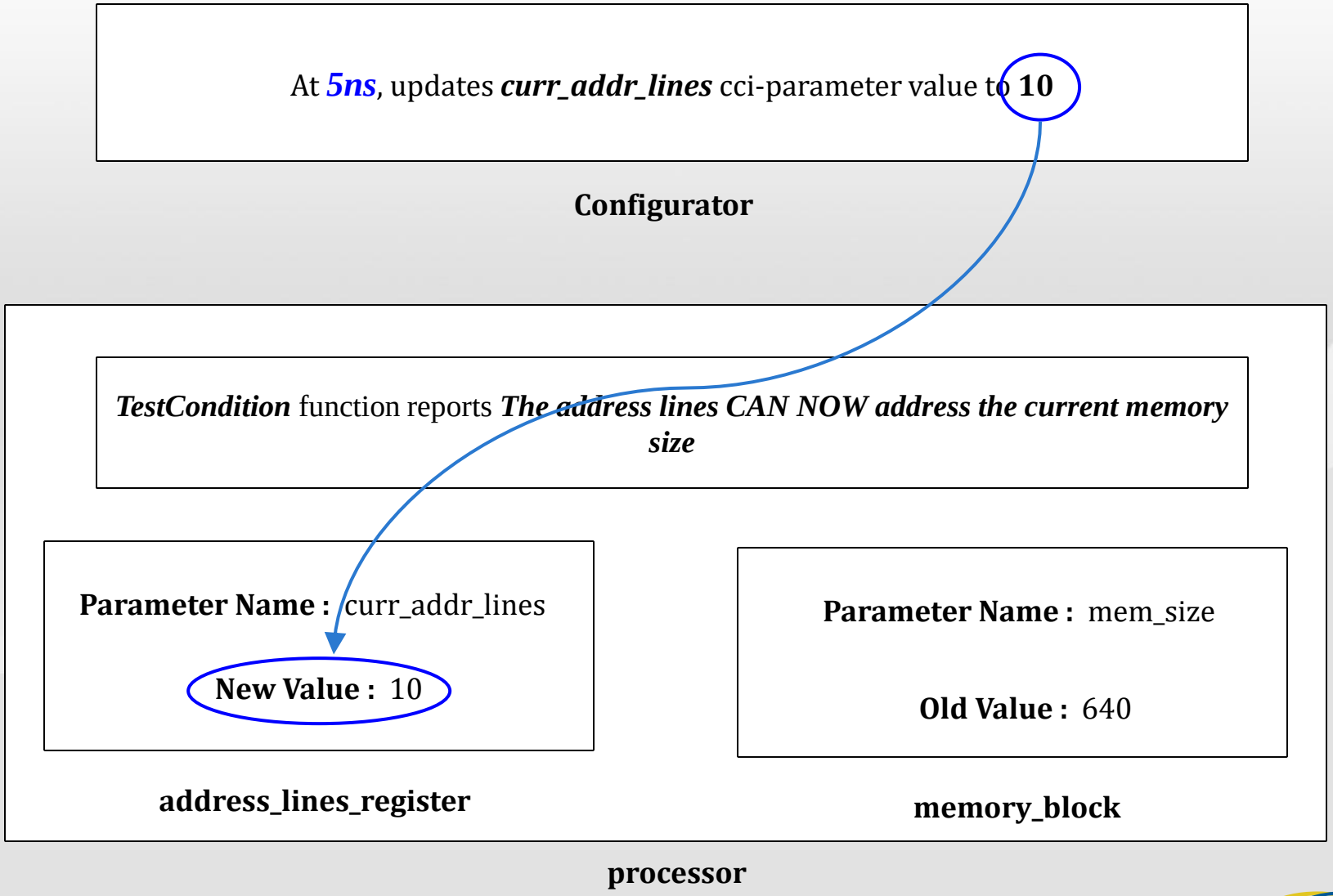
Example Illustration (3)



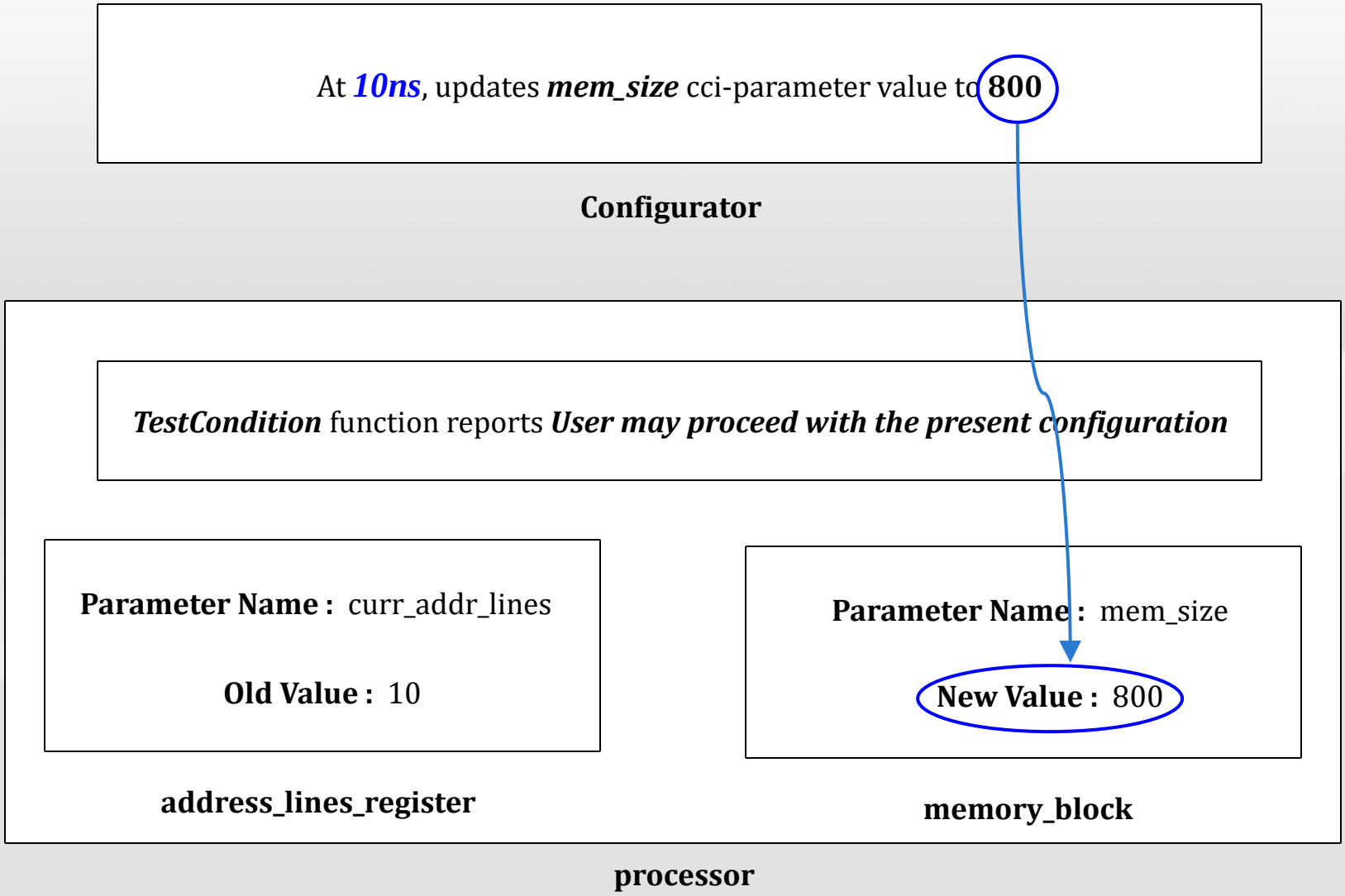
Example Illustration (4)



Example Illustration (5)



Example Illustration (6)



Expected Output
(ex05_Inter_Parameter_Value_Constraint.log)

```
SystemC Simulation
Info: sc_main: [MAIN] : In this example, the following is condition is verified
Info: sc_main: [MAIN] : x <= 2^n - 1
Info: sc_main: [MAIN] : where, 'x' : value of 'mem_block_size' (Memory Block size), and
Info: sc_main: [MAIN] : 'n' : total number of address lines - 'curr_addr_lines'
Info: processor.addr_lines_mod: @0 s, [ADDR_LINES_REG C_TOR] : Default Address Lines : 9
Info: processor.memory_block: @0 s, [MEMORY_BLOCK C_TOR] : Default Memory Size : 500
Info: processor: @0 s, [PROCESSOR fn] : User may proceed with the present configuration
Info: sc_main: Begin Simulation.
Info: param_cfgr: @0 s, @ 0 s
Info: param_cfgr: @0 s, [CFGR] : Changing the 'mem_size' to 640
Info: processor: @0 s, [PROCESSOR mem_block_post_wr_cb] : Parameter Name :
processor.memory_block.mem_size  Parameter Value : 640
Info: processor: @0 s, [PROCESSOR mem_block_post_wr_cb] : Parameter Name :
processor.addr_lines_mod.curr_addr_lines Parameter Value : 9
Info: processor: @0 s, [PROCESSOR fn] : The address lines cannot address the current memory size
```

Cont’d

```
Info: param_cfgr: @5 ns, @ 5 ns

Info: param_cfgr: @5 ns, [CFGR] : Modify the 'curr_addr_lines' to 10

Info: processor: @5 ns, [PROCESSOR addr_lines_post_wr_cb] : Parameter Name :
processor.addr_lines_mod.curr_addr_lines   Parameter Value : 10

Info: processor: @5 ns, [PROCESSOR addr_lines_post_wr_cb] : Parameter Name :
processor.memory_block.mem_size    Parameter Value : 640

Info: processor: @5 ns, [PROCESSOR fn] : The number of address lines can now address the current memory
size

Info: param_cfgr: @10 ns, @ 10 ns

Info: param_cfgr: @10 ns, [CFGR] : Changing the 'mem_size' to 800

Info: processor: @10 ns, [PROCESSOR mem_block_post_wr_cb] : Parameter Name :
processor.memory_block.mem_size    Parameter Value : 800

Info: processor: @10 ns, [PROCESSOR mem_block_post_wr_cb] : Parameter Name :
processor.addr_lines_mod.curr_addr_lines   Parameter Value : 10

Info: processor: @10 ns, [PROCESSOR fn] : User may proceed with the present configuration

Info: sc_main: End Simulation.
```